Marie Biernacki

COMP-SCI 303 Data Structures

Professor Shah

Assignment 03 Report

**PROGRAM EXECUTION:**

* **IN MAIN.CPP**
  + include Queue.h and Functions.h
  + create and initialize code that tests Queue and Functions functionality
    - PART 1: should test the following functions for Queue:
      * push()
      * empty()
      * move\_to\_rear()
      * displayQueue()
    - PART 2: should test the following functions for Functions
      * linear\_search()
    - PART 3: should test the following functions for Queue:
      * insertion\_sort()
      * displayQueue()
* **IN QUEUE.H**
  + **create the following template structs**
    - Item\_Type
      * define data variable, create the Item\_Type constructor
    - Node
      * define nodeData, pointer to the next node and the Node constructor
  + **create the Queue class as a template class**
    - **protected members:** 
      * head pointer, tail pointer, and num\_items
    - **public members:**
      * default constructor, constructor passing Item\_Type, and destructor
      * **declare and define the following functions:**
        + void push(const Item\_Type<T>& item)

adds node to the end of the queue

* + - * + void pop()

removes the first node of the queue

* + - * + Item\_Type<T>& front()

returns first element in the queue

* + - * + int size()

returns current size of the queue

* + - * + bool empty()

returns true if queue is empty

* + - * + void displayQueue()

displays all the elements in the queue using only size, front, pop, and push functions

* + - * + void move\_to\_rear()

takes element at the front of the queue, moves it to the rear of the queue

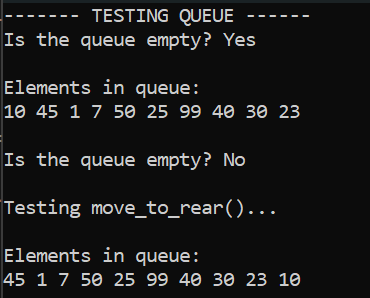
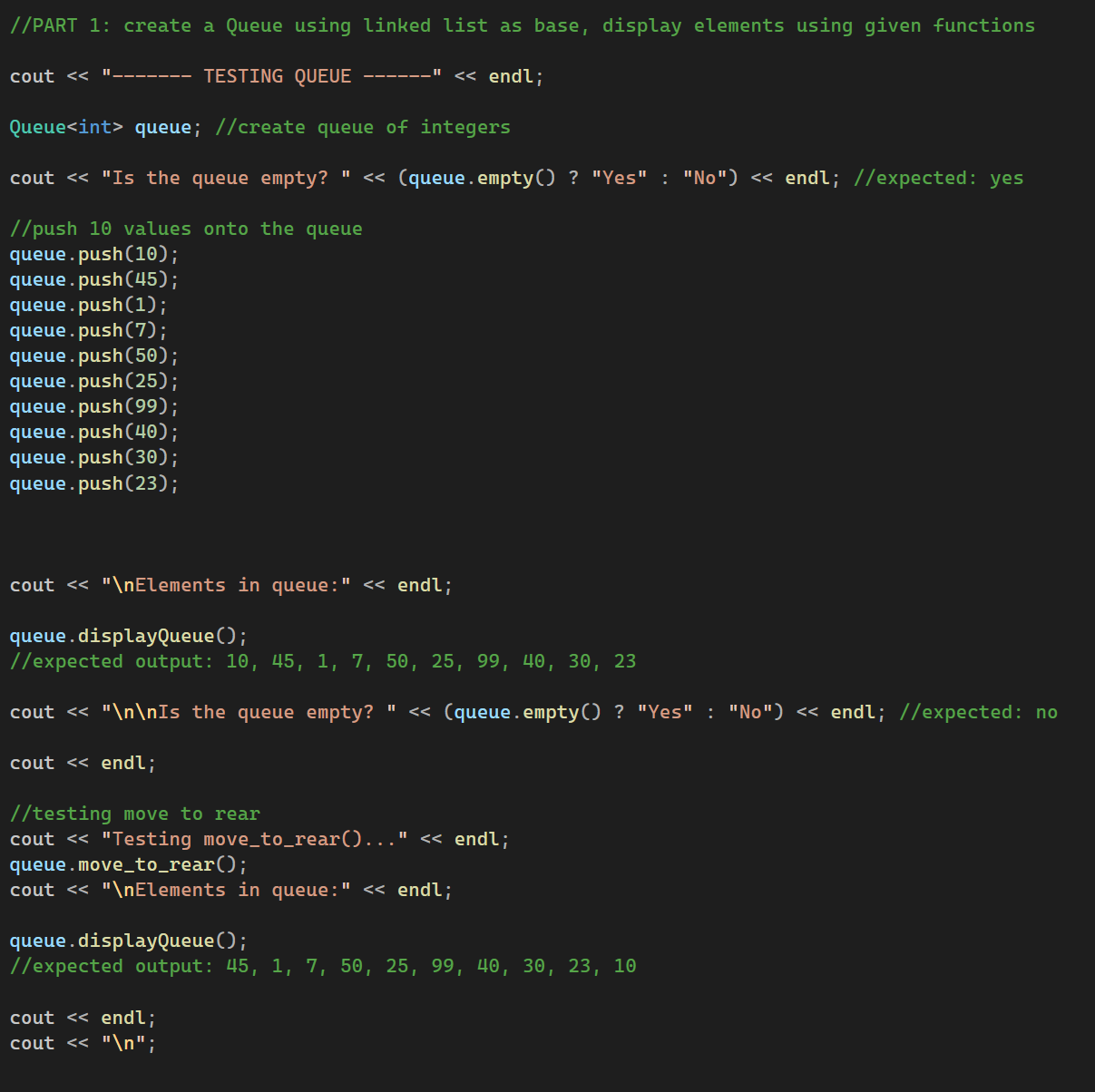
* + - * + void insertion\_sort()

sorts queue in ascending order

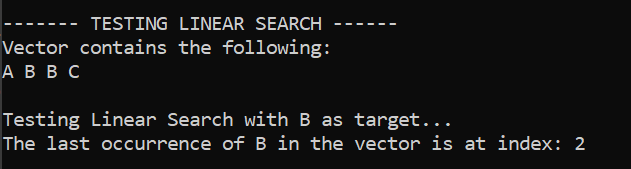
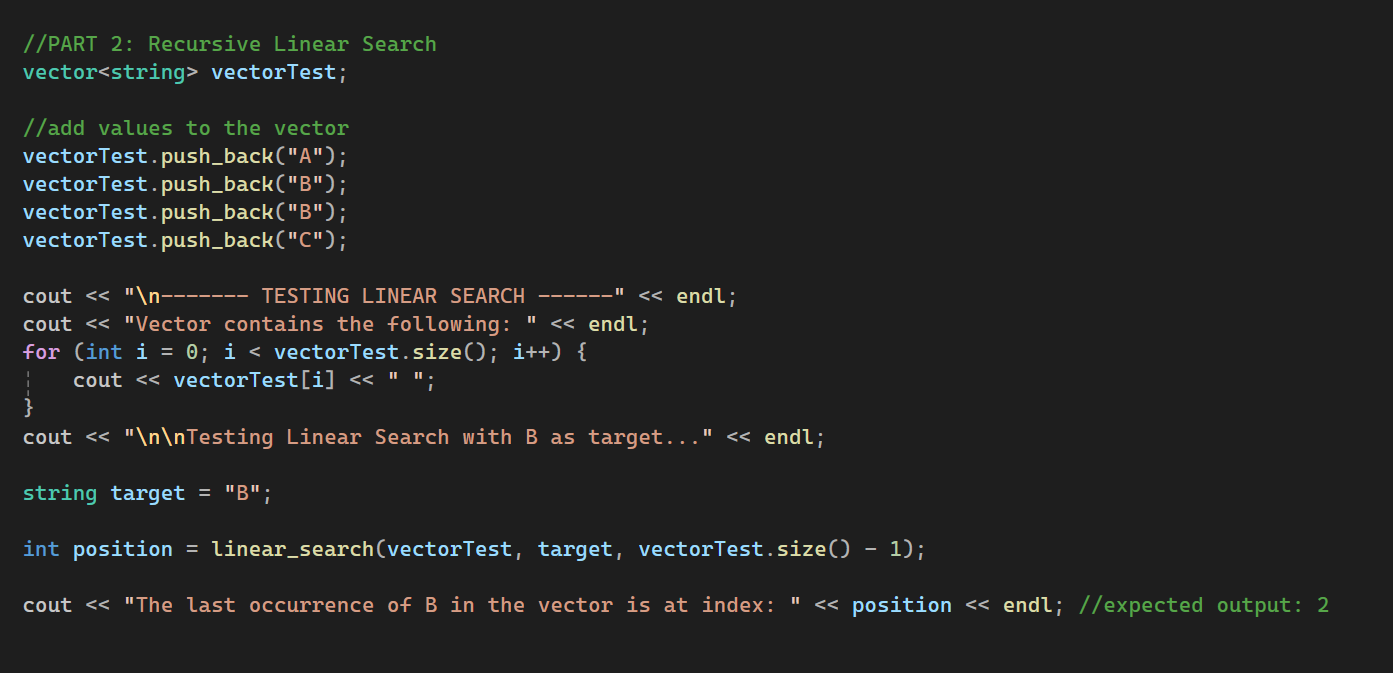
* **IN FUNCTIONS.H**
  + include standard vector library to use for functionality
  + define the linear\_search function as a template function
  + int linear\_search(const vector<Item\_Type>& items, Item\_Type& target, size\_t index)
    - finds the index of last occurrence of target in the vector using recursive linear search

**OUTPUT AND CODE SCREENSHOTS:**

*Part 1: Queue Data Structure with 10 Integers, Display Elements, Move to Rear*



*Part 2: Recursive Linear Search on Vector*



*Part 3: Insertion Sort on Queue*

